

LISTING OF THE CLAIMS:

1. (Currently Amended) A method of processing a text file in a computer application, comprising the steps:

forming a template having literal fragments of the text file;

providing a macro class to map data from the text file to the computer application;

embedding in the template a position pointer to the macro class; and

using the template as an overlay to parse the text file into segments having data, or as a prototype to generate a segment of an output file;

said using step including the steps of:

i) reaching said pointer in said template, and

ii) when said pointer is reached, using said pointer to invoke said macro class and using said macro class to map data from one of the segments of the text file to the computer application; and

iii) said macro class then invoking a next template to further process the text file.

2. (Original) A method according to Claim 1, wherein the macro class reads in a segment of the text file and uses the segment to initiate application update processing.

3. (Original) A method according to Claim 1, wherein the macro class derives data from the application and formats it into the text file.

4. (Currently Amended) A method according to Claim 1, wherein the step of invoking a next template includes the step of the macro class ~~derives~~ deriving a template name from the invoking template and ~~uses~~ using that name to invoke a next template to further process the text file.

5. (Original) A method according to Claim 1, further comprising the step of providing an interface controller to prevent structure clashes by placing text data into appropriate places in a complex object structure as the text file is processed.

6. (Currently Amended) A system for processing a text file in a computer application, comprising:

means ~~for~~ forming a template ~~from~~ having literal fragments of the text file;

means forming a macro class to map data from the text file to the computer application, wherein a pointer to the macro class is embedded in the template;

means for using the template as an overlay to parse the text file into segments having data, or as a prototype to generate a segment of an output file;

said means for using including

- i) means for using said pointer, when said pointer is reached in said template, to invoke said macros class, ~~and~~
- ii) means to use said macro class to map data from one of the segments of the text file to the computer application;
- iii) means to use the macro class to invoke a next template to further process the text file.

7. (Original) A system according to Claim 6, wherein the macro class reads in a segment of the text file and uses the segment to initiate application update processing.

8. (Original) A system according to Claim 6, wherein the macro class derives data from the application and formats it into the text file.

9. (Original) A system according to Claim 6, further comprising an interface controller to prevent structure clashes by placing text data into appropriate places in a complex object structure as the text file is processed.

10. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for processing a text file in a computer application, said method steps comprising:

forming a template having literal fragments of the text file

providing a macros class to map data from the text file to the computer application;

embedding in the template a pointer to the ~~means~~ macro class; and

using the template as an overlay to parse the text file into segments having data, or as a prototype to generate a segment of an output file;

said using step including the steps of:

- i) reaching said pointer in said template, ~~and~~
- ii) when said pointer is reached, using said pointer to invoke said macro class and using said macro class to map data from one of the segments of the text file to the computer application;
- iii) said macro class then invoking a next template to further process the text file.

11. (Original) A program storage device according to Claim 10, wherein the macro class reads in a segment of the text file and uses the segment to initiate application update processing.

12. (Original) A program storage device according to Claim 10, wherein the macro class derives data from the application and formats it into the text file.

13. (Original) A program storage device according to Claim 10, wherein said method steps further comprise the step of providing an interface controller to prevent structure clashes by

placing text data into appropriate places in a complex object structure as the text file is processed.

14. (Previously Presented) A method according to Claim 1, wherein:

the step of using said macro class includes the step of, said macro class using said name to invoke said another template to further process the text file.

15. (Previously Presented) A system according to Claim 6, wherein:

when the macro class is invoked, a name for another template is passed to the macro class from said template; and

said macro class uses said name to invoke said another template to further process the text file;

16. (Currently Amended) A program storage device according to Claim 10, wherein

when the macro class is invoked, a name for another template is passed to the macro class from said template; and

the step of using said macro class includes the steps of, said macro class using said means name to invoke said another template to further process the text file.